Training Kohonen Feature Maps in different Topologies: an Analysis using Genetic Algorithms

Daniel Polani and Thomas Uthmann
Institut für Informatik
Johannes Gutenberg-Universität
Staudingerweg 9, D-6500 Mainz, Germany
polani@informatik.mathematik.uni-mainz.de
uthmann@informatik.mathematik.uni-mainz.de

Abstract

In the following paper a Genetic Algorithm is applied to improve the topology of Kohonen Feature Maps with respect to a simple fitness function measuring the quality of the trained net. The training of a Feature Map is an adaptation process whose effectivity depends on the topology chosen for the net. We show several results, where the Genetic Algorithm is used as a tool to study the effectivity of the adaptation of Kohonen Nets to input signal spaces of toroidal and Möbius topology.

1 INTRODUCTION

Genetic Algorithms (GA) are often applied to large problem domains where the domain structure is not very well known and a priori-knowledge is scarce [5], [6]. The claim is that GAs yield an efficient exploration of the search space in question. Thus GAs could be considered as valuable tool for analysis of complex behaviour [11].

The boost of Neural Network (NN) research has been caused by observation of similar properties of NNs in nondiscrete, continuous domains, especially in domains with strong geometrical structure. In the latter case successful modeling has often been achieved by usage of *Kohonen Feature Maps* [8].

Recent interest in a combination of both paradigms arises from the wish to make use of the combinatorial power of a GA to enhance the flexibility of adaptation of the Networks. Behind these motivation lie, of course, the biological roots both approaches have in common.

A main application of GAs to NNs has been the usage of the GA as a learning rule together with or instead of Backpropagation or related algorithms to train the weights of Feed-Forward-Networks [12]. They have also been used for searching optimal learning rules [3] and efficient net structure and topology [2], [12]. It has been pointed out that a main target of combined GA-NN research should be the search for a non-explicit genotype-phenotype coupling [1]; a direct coding often proves inappropriate and may lead to an unnecessarily large GA search space.

With their particular ability to adapt to simply and uniformly structured event spaces Kohonen Feature Maps need simple and regular net topologies for these problem domains; thus they offer the opportunity to study genotype-phenotype-mappings where the phenotypes are highly regular nets with a high degree of redundancy. On the other hand, this enables us to study the assumptions that rule the design of a Kohonen Net which is applied to a given problem.

In [9] a scheme for coupling GAs and the Kohonen model has been introduced and first results concerning the adaptation abilities of the Feature Maps to the unit square with trivial topology have been presented. For the sake of self-containedness we will first recapitulate the required extensions of the Kohonen model, the genotype-phenotype transcription rule and the fitness function, which has been used in the simulations.

The following parts of the paper will deal with new results. We will analytically derive values of the fitness function for several simple configurations to obtain a criterion for the fitness values to expect from the trained nets. The results of applying a GA to study the adaptation of Kohonen nets to input signal spaces of toroidal and Möbius topologies over broad parameter ranges and using different activation profiles are presented and discussed.

2 THE COUPLING SCHEME

2.1 THE KOHONEN FEATURE MAP

The Kohonen Feature Map can be considered as an undirected graph $G = (V, E)$, where a weight vector $w_v \in \mathcal{R} \subseteq \mathbb{R}^m$ is associated with every vertex $v \in V$. On the set of vertices (or *neurons*) of G we have the canonical metric d_G ($d_G(v, v') = 1$ for $v, v' \in G$ iff they

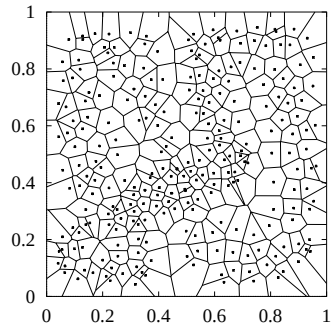

Figure 1: Receptive fields of net from Fig. 4.(a)

are connected by an edge) and on $\mathcal{R}$ a metric $d_{\mathcal{R}}$. If $\mathcal{R}$ is a simple hypercube $[0,1]^m$, $d_{\mathcal{R}}$ is typically chosen as the Euclidean distance. The *receptive field* F_v of a neuron $v \in V$ is defined by

$$F_v = \{w | w \in \mathcal{R}, \forall v' \in V, v' \neq v : d_{\mathcal{R}}(w, w_v) < d_{\mathcal{R}}(w, w_{v'})\},$$

i.e. by all weight vectors w, for which w_v is the nearest weight. When $d_{\mathcal{R}}$ is chosen as the Euclidean distance, F_v with $v \in V$ yields just the well-known Voronoi tesselation [7]. The receptive fields F_v yield a partition of $\mathcal{R}$ (we ignore the boundaries of F_v). As an example, in Fig. 1 we give the partition of $\mathcal{R}$ into receptive fields for the net given in Fig. 4.(a).

The Kohonen training procedure can then be applied as follows to a general topology [8], [10]: In every training step first a vector $x \in \mathcal{R}$ is presented (e.g. according to a probability distribution). Then the neuron v^* is chosen, for which $x \in F_{v^*}$. In the simple Euclidean case the correction of the weights is then computed according to

$$w_v(t+1) = w_v(t) + \alpha(t) \cdot h_t(d_G(v^*, v)) \cdot [x(t) - w_v(t)] \quad (1)$$

for every neuron v. The correction depends on the learning rate $\alpha(t)$ and on the activation profile h_t. The latter determines how strongly the weights of a neuron are corrected depending on its distance $d_G(v^*, v)$ from the maximally activated neuron v^*.

Typical choices of the net topology include arrangement of the neurons on a regular lattice and connection of immediate neighbours. In other network architectures derived from the Kohonen model network topology changes during the training process (e.g. as in [4]). In both cases the network topology is determined by local criteria, in the explicit (a priori fixed topology) as well as in the variable topology case.

We will now focus our interest to the case where the topology is fixed and explicitly determined by a GA, without however using an explicit a priori assumption about the local structure of the Kohonen graph.

2.2 COUPLING OF GA AND KOHONEN FEATURE MAP: OVERALL STRUCTURE

Fig. 2 shows the general structure of the algorithm. The GA creates by mutation and recombination chromosomes $\in \{0,1\}^q$. Every such genotype defines the topology of a Kohonen Net via a transcription rule. A Kohonen training as described above is performed and the trained net is evaluated by a quality function which determines the fitness as required by the GA.

2.2.1 Transcription Rule

We will give a short description of the transcription rule used in our simulations. The transcription rule yields a net with a unique topology for every chromosome. The rule is applied to chromosomes consisting of 2 bytes + `n` double-bytes (1 double-byte = 2 bytes = 16 bits):

k - 1	maxsteps/16	a_0	b_0	a_1	b_1	$\cdots$	a_{n-1}	b_{n-1}

If `k` is the number of neurons in G then the first byte of the genotype contains the number `k-1`, thus the existence of at least one neuron in the net is guaranteed. The second one multiplied by 16 yields the maximum number of transcription steps `maxsteps` that may be done before the procedure has to stop. These two "header bytes" are followed by `n` double-bytes, each of which describes one transcription step. The neurons are numbered from $0 \ldots k-1$. Beginning with step $i = 0$ the i-th transcription step consists of moving from the current neuron v to neuron $v + a_{(i \bmod n)} \bmod k$ and connecting this one with $(v + a_{(i \bmod n)} + b_{(i \bmod n)}) \bmod k$. Whenever the rule tries to connect two already connected neurons or tries to connect a neuron with itself, the algorithm stops. A more detailed description can be found in [9].

As the topology is given as a result of the transcription rule, the net is trained for a fixed number of steps according to the Kohonen procedure as described in section 2.1. Thus the genotype-phenotype mapping consists of two phases: a deterministic one describing the transcription of a chromosome to a net topology; and a (in general) nondeterministic one, during which the network weights are trained to adapt themselves to the input signal space. The full phenotypic expression is a result of the combination of "genetic" and "environmental" factors. One note should be added here: `maxsteps` may, but need not influence the net topology. It sets only an upper bound on the number of connections created, which need not be exploited by the transcription algorithm.

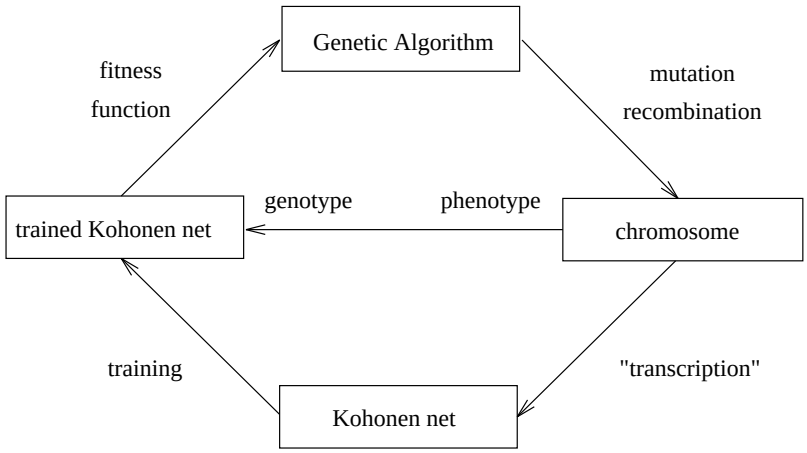

Figure 2: Interaction between the GA and the Kohonen Feature Map

The contents of the double-bytes define the "rhythm" by which new edges are created between neurons. The first byte of the chromosome has a strong influence on the construction of the net since it not only determines the number of neurons k but also the semantics of the double-bytes via the modulo operation taking place during the transcription.

When a double-byte m is inherited by an offspring (assuming it is not disrupted due to crossover) it only retains its "rhythmic" meaning if in the offspring the modulo values of a_m and b_m with respect to k are the same like in the parent; this is the case if parent and offspring have the same first byte (thus the same k) and also for the sporadic values of k', for which $a_m \bmod k' = a_m \bmod k$ (and $b_m \bmod k' = b_m \bmod k$, resp.). But the probability that several inherited double bytes will keep their semantics sinks rapidly if their first bytes are not identical. A tight coupling between a double byte and the chromosome's first byte would be needed to insure that the meaning of the double-byte is retained.

Since it turned out that the GA tried to achieve large values for the first byte $(k-1)$ to increase neuron density and thus the value of the quality function Q (see below) during the runs, k assumed values in the range above about 230 for the majority of the population, often after less than 100 generations. Therefore, after some generations, only a few distinct values for k were left, dividing the population into "families"; within such a family, the semantics of a double-byte did not change by inheritance. This results in a certain probability that a double-byte carrying a valuable piece of topology information can be transferred from a parent to its offspring without losing its meaning.

When double-bytes are disrupted by crossover there is no direct interpretation of what kind of information is transferred by the "sub-double-byte" elements. We choose however not to disallow disruption within a double-byte to enable genetic diversity.

2.2.2 Quality function

To apply a GA to evaluate the chromosomes, a quality or fitness function is needed to measure the performance of the trained net. The quality function used here is given by

$$Q(G) = 1/\sum_{i=1}^{q}(w^*(x_i) - x_i)^2$$

where $x_i \in \mathcal{R}$, $i = 1, \ldots, q$ are randomly chosen and $w^*(x_i) := w_{v^*(x_i)}$ if $v^*(x_i)$ is the neuron activated by the signal x_i. Q is a measure of the average distance of an input signal from the corresponding neuron, evaluating how efficiently the neurons' receptive fields cover $\mathcal{R}$. It is therefore an indirect measure of the equidistribution of neuron weights in $\mathcal{R}$. As the weights are more evenly distributed, Q yields a higher value, since the average distances of a signal to its activated neuron is lowered in this case.

3 ANALYSIS OF THE QUALITY FUNCTION

We will now estimate the value of $Q(G)$ for three exemplary distributions of weights on $\mathcal{R} = [0,1]^2$. The weights are distributed 1. on a square lattice, 2. on a hexagonal lattice (which is the densest 2-dimensional spherical packing [7]) and 3. randomly on $\mathcal{R}$. In all cases the number of neurons (weights) k is 256 and q is 100. To simplify the computation we assume periodic boundary conditions on $\mathcal{R}$ (thus ignoring boundary effects which otherwise would arise). To compute Q the probability density $p(r)$ that a sample point x_i has distance r to $w^*(x_i)$ has to be calculated. For the regular lattices this can be done by considering the receptive fields (Voronoi cells) for the weights w_v: in the

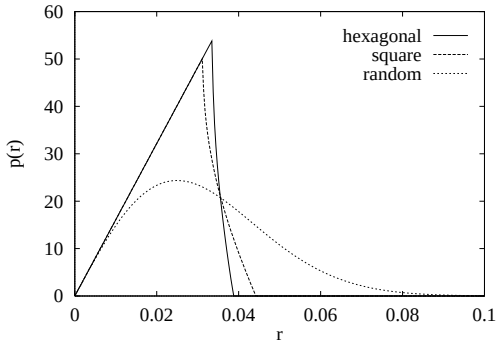

Figure 3: The distribution $p(r)$

square lattice these are squares with w_v in the center. In the hexagonal lattice the Voronoi cells are regular hexagons of the same size with w_v in the center. The area of the receptive fields are the same for different neurons[1]. A randomly chosen $x \in \mathcal{R}$ falls into one of the cells with probability 1. Thus the problem of finding $p(r)$ reduces to computing the probability density that an x randomly chosen from a square or hexagonal cell (for the square and the hexagonal lattice, resp.) has distance r from the center of the cell. The results of the calculation, which is straightforward, are plotted in Fig. 3.

To compute $p(r)$ for a randomly distributed set of weights w we simplify the computation by assuming that k is so large that $p(r)$ can be neglected for $r > 1/2$. We consider the set $W = \{w_i | i = 0 \ldots k-1, w_i \in \mathcal{R}\}$, where the w_i are chosen randomly and a randomly chosen $x \in \mathcal{R}$. The disk with radius r around x is denoted by $b(r)$. Then $p(r)$ can be obtained by computing the probability $P(r, \Delta r)$ that $b(r) \cap W = \emptyset$, but $b(r + \Delta r) \cap W \neq \emptyset$ and letting $p(r) = \lim_{\Delta r \to 0} P(r, \Delta r)/\Delta r$. Its graph is shown in Fig. 3.

The expectation value for $Q(G)$ is given by the expression $1/(q \cdot \langle r^2 p(r) \rangle)$, where $\langle r^2 p(r) \rangle = \int r^2 p(r) dr$. This yields a value of 15.96 for the hexagonal lattice, of 15.36 for the square lattice and of 8.07 for the random distributed weights. These values will serve as a scale for the adaptation quality which can be expected from the trained nets.

4 EXTENSIONS TO NONTRIVIAL TOPOLOGIES

In [9] a GA has been used to adapt Kohonen net topologies to $\mathcal{R} = [0,1]^2$ using the Euclidian distance as $d_{\mathcal{R}}$. It was observed that training times of 3000 learning steps would favour regular nets with a locally "flat" topology, which, on the other hand, show some difficulties in global adaptation to $\mathcal{R}$. Fig. 4.(a) shows the result of such a run. With the transcription rule from section 2.2.1 every net topology can be realized, provided that the chromosome length is sufficiently large. For shorter chromosome length, however, the symmetry of the nets is higher, suggesting that the adaptation to a signal input space $\mathcal{R}$ with a higher degree of symmetry than the simple square would lead to better adaptation of the Kohonen Feature Maps. For this reason in the simulations in the current paper input space $\mathcal{R}$ has been again chosen as $[0,1]^2$, but with a toroidal and Möbius topology, respectively, which have both a continuous symmetry group. Choice of a different topology implies choice of a different metric $d_{\mathcal{R}}$ and of a correction (learning) rule different of (1).

One obtains a *toroidal topology* on the unit square by identifying the left and right edge and the top and bottom edge of the square, respectively. Toroidal topology on the unit square is equivalent to periodic boundary conditions. This topology induces a different metric and learning rule, which are derived from the affine plane with the Euclidean metric, and which we will now briefly sketch. We wish to determine $d_{\mathcal{R}}(x, w)$ for arbitrary $x, w \in \mathcal{R}$. We note that every element z of $\mathcal{R} = [0,1]^2$ is in fact a representative for the equivalence class $[z] = \{z + ie_1 + je_2 | i, j \in \mathbb{Z}\} \subseteq \mathbb{R}^2$, where $e_1 = (1,0)^T$ and $e_2 = (0,1)^T$. Then

$$d_{\mathcal{R}}(x, w) := \min_{x' \in [x], w' \in [w]} d_{\text{euc}}(x', w'),$$

where d_{euc} is the usual Euclidean metric and the minimum value is obviously assumed. For every vertex v the correction for $w_v(t)$ analogous to (1) is then calculated by choosing $x'_v \in [x(t)]$ and $w'_v \in [w_v(t)]$, for which the above minimum distance is assumed and then applying (1) to x'_v and w'_v:

$$w'_v(t+1) = w'_v(t) + \alpha(t) \cdot h_t(d_G(v^*, v)) \cdot [x'_v(t) - w'_v(t)] \quad (2)$$

The new weight $w_v(t+1)$ is then chosen as the representative of $[w'_v(t+1)]$ which lies in the unit square. $w_v(t+1)$ is unique with probability 1.

The *Möbius topology* is obtained by identifying the left and right edge, the latter of which is twisted; thus e.g. (0,1) is equivalent to (1,0) and (1,1) is equivalent to (0,0). $d_{\mathcal{R}}$ and the learning rule are calculated similarly to the toroidal case, taking into account the twist when determining the equivalence classes.

In the next section we will discuss the simulation results. The plots essentially show the standard Kohonen representation of the nets, where the weights $w_v, w_{v'}$ for adjacent neurons v and v' are connected by a line. In the plots showing toroidal and Möbius topology adaptations the standard representation is altered insofar as the connections are drawn not necessarily

[1] Here we make the simplifying assumption that the gaps created by fitting the lattices into $\mathcal{R}$ can be neglected when k is not too small

between $w_v, w_{v'} \in \mathcal{R}$, but between w_v and the nearest representative of $[w_{v'}]$ and between $w_{v'}$ and the nearest representative of $[w_v]$; this to clarify the topological coherence.

5 SIMULATION RESULTS

In the simulations a GA with roulette wheel selection was applied to the above described fitness function Q, as recombination 2-point crossover was used. The population consisted of 30 chromosomes initialized with random binary strings. The mutation is characterized by two parameters p_{G} and p_{m}. The probability p_{G} that a mutation "sweep" is performed on the population of the current generation is 30%. $p_{\mathrm{m}} = 10\%$ is the probability that succeeding mutations take place in a chromosome, i.e. the probability that n succeeding bit flips at random independent positions in a chromosome take place is $p_{\mathrm{m}}^n \cdot (1 - p_{\mathrm{m}})$. The best chromosome was never subjected to mutation or substitution by offspring.

To calculate the fitness function the procedure described in 2.2.1 was used to construct a net topology from the given chromosome. Before training, the neuron weights were randomly distributed in $[0.45, 0.55]^2$. The Kohonen training with a sequence of random vectors $\in [0,1]^2$ was then performed. Afterwards the quality function was calculated for a sample of $q = 100$ randomly chosen test vectors. The random vector sequences used were the same for every chromosome for the sake of reproducibility. Thus the genotype-phenotype mapping was indeed a deterministic procedure, yielding the same trained net for the same chromosome and the same quality value was obtained for the same net. In the more general case, a random training sequence and sample would be selected and the value of the quality function would depend stochastically on the chromosome. We used $\alpha(t) = 0.5/(1 + 0.004t)$ as learning rate and two different activation profiles $h^{(1)}$ and $h^{(2)}$. $h_t^{(j)}(v, v')$ takes the value 1, iff $d_G(v, v') \leq R^{(j)}(t)$, otherwise the value 0. $R^{(1)}(t) = 2$ for $0 \leq t < 2000$ and $= 1$ otherwise. $R^{(2)}(t) = 3$ for $0 \leq t < 1000$, $= 2$ for $1000 \leq t < 2000$ and 1 otherwise. R can be regarded as a characteristic activation range, which is smaller for $h^{(1)}$ than for $h^{(2)}$.

In the following we will show the results of the GA runs. All runs extended to 4000 generations and in Fig. 4.(e)-(p) we show the best phenotype of the corresponding run. A description of the runs is given in the text. As our aim is to study the "morphology" of the resulting Kohonen maps we will show selected results showing typical and extreme behaviour. A certain bandwidth of results is given by variation of mutation parameters, activation profiles and chromosome lengths.

Fig. 4.(a) shows a run from [9], where the GA-Kohonen coupling has been applied to the simple unit square with simple topology. The chromosomes have had $n = 5$ double-bytes and the resulting quality after 500 generations has been 9.04 with the activation profile $h^{(2)}$.

The Figures 4.(b)-(l) display results as obtained by using toroidal topology, results with Möbius topology are shown in Fig. 4.(m)-(p).

Figs. 4.(b)-(e) show the best phenotype of generation 0, 11, 1992 and 2440 with $n = 3$, activation profile $h^{(1)}$, where the qualities attained are 11.57, 13.18, 14.28 and 16.72 respectively. One notes a tendency to increased order in the trained net towards higher fitness function values. In Fig. 4.(b) and (d) the formation of weight clusters can be observed by which certain regions of $\mathcal{R}$ are shared by more neurons than others. This leads to a unnecessary redundancy that usually causes loss of the quality value. Comparison of Figs. 4.(c) and (d), on the other hand, show a deviation from this behaviour.

The amount of connection crossings in $\mathcal{R}$ (measuring the "non-flatness" of the net) decreases for later generations, but even the last net is not completely flat – the reader is invited to take a close look at Fig. 4.(e), which will show crossings systematically distributed over the net. One may observe that the quality value found by the GA is better than the value expected of a hexagonal lattice as calculated in section 3. One should note, however, that the GA of course is able to exploit certain features and random fluctuations of our - in a sense - very specialized case, as always the same training and sample values are used. This effect, too, renders it possible that in Fig. 4.(d) the quality value is higher than in Fig. 4.(c) despite the cluster formation in (d).

Our a priori expectation was that nets with a flat topology (e.g. without connection crossings) and hexagonal structure would have been favoured. The GA runs however show the existence of nets with non-flat topologies with the ability to be trained to quality values lying even higher than the expectation value for the optimal flat topology (for our particular random sample). Thus we notice that optimally or nearly optimally trainable topologies lie distributed over different regions in "topology space".

Whether the regular but nonflat lattice found above is a true optimum or only a result of the specialization to a fixed training set and test sample, is an open question.

To study their influence we performed runs with different mutation parameters. Fig. 4.(f) shows the result of a different GA run with the same chromosome length ($n = 3$), but lower mutation rates of $p_{\mathrm{G}} = 5\%$ and $p_{\mathrm{m}} = 10\%$; the untypically low value of the quality

function of 13.67 is attained as early as the generation 213 and the phenotype exhibits a high degree of connection crossings. Structure and fitness value are similar to the net shown in Fig. 4.(c). Inspection of the final population revealed a high degree of inbreeding. This is a typical phenomenon found when using small mutation rates. Our choice of $p_G = 30\%$ and $p_m = 10\%$ for the other runs presented here have proven to guarantee a sufficient degree of gene diversity.

Fig. 4.(g) shows the best net of a run with $h^{(2)}$, i.e. with a wider activation range. The best solution (quality 16.12) has been found in the 3991th generation and the net topology enables a completely flat embedding; on the other hand the completely trained phenotype does not exploit this property. Instead, its adaptation takes advantage of the particular set of sampling points.

One could expect that $h^{(2)}$ would favour flat (planar) net topologies compared to $h^{(1)}$. In Fig. 4.(h), however, the best phenotype of a run with $h^{(2)}$ is shown (quality 14.81, found after 658 generations). There large regions with a flat net topology are found, although lacking the correct global behaviour.

The net of Fig. 4.(i) is the strongest phenotype that was found in all runs with a quality of 17.20 and which was obtained using $h^{(1)}$ in the 460th generation. The clearly nonflat character of the net is remarkable. Again, the reader is invited to inspect the regions around (0.4, 0.5) and (0.9, 0.9), where inhomogeneous structures can be made out.

The nets presented in the following figures have been using the activation profile $h^{(2)}$. To study the influence of longer chromosomes we performed runs with $n > 3$. Fig. 4.(j) shows the fittest phenotype of a run using chromosomes with $n = 5$. A complex, nonflat net results after 1175 generations with a quality value of 16.93. In general, one observes that longer chromosomes lead to more complex structures: good topologies are harder to find and the obtained quality values are worse (the best obtained quality for $n = 7$ is 15.05 in our runs). Whether n is even or odd does not affect the observations considerably.

Figs. 4.(k) and (l) show two examples of nets with same topology but with different weight distributions and qualities (14.47 vs. 15.44). Although the initial weights and the training data are the same for both nets they are distributed differently on the neurons because the weights for the neurons are initialized for the neurons $0 \ldots k-1$ in turn, but the way they are connected to give a hexagonal lattice is different.

In general, repeated training runs with the same net topology but with different random training sequences yields different quality values for the trained nets and a probability distribution of obtaining a net with a given quality value. After 500 independent training runs with the same topology a marked distribution of quality values can be observed. The distributions typically exhibit one or two maxima (not shown here). In the case of two maxima the maximum at the lower quality values was often caused by nets which were unable to unfold completely during the training process although their topology could have been embedded properly into signal space. The distribution showed that the quality values may extend from values of 2.0 to over 17.0 for better topologies or to 9.0 or even less for weaker topologies. However, the phenotypes found by the GA range at the upper limit of these distributions – they thus represent the best or the nearly best training result of a net with a given topology. This permits an estimate of the maximum quality that *can* be achieved with a given topology. It does not ensure that a topology *will* yield such high quality values after training. If one were interested in creating topologies which would give high quality values after training with a certain reliability, then some kind of average over several training runs would have to be done.

In Fig. 4.(m) we show the result of a run where the system had to adapt to a Möbius topology. n was chosen as 3, and after 1048 generations, the run's maximum fitness of 15.60 was found, which also was the highest fitness found in all Möbius runs. In Fig. 4.(n)-(p) the nets created during a run with $n = 7$ are plotted for the Möbius topology. In Fig. 4.(n) the best phenotype of the initial generation is shown with a quality of 6.83, Fig. 4.(o) gives the best phenotype of generation 37 (quality 10.71) and Fig. 4.(p) the best phenotype of the run (quality 11.85) attained after 1777 generations.

We see a strong tendency to formation of order and reduction of connection crossings in the Möbius runs, of which the latter is a typical representative; one should note that especially the long chromosomes ($n = 7$) support higher connectivity, and therefore the run demonstrates clearly that weaker connectivity is favourable with our fitness function on the Möbius strip.

The selection pressure to reduce connection crossings seems stronger on the Möbius strip than in the toroidal case (which can be observed equally in runs with $n = 5$ or $n = 7$). In the toroidal case it has proven easier to obtain high fitness values. We attribute this to the symmetry induced by the transcription rule and the high symmetry of the torus. On the other hand, the smaller symmetry group of the Möbius strip seems to force the algorithm to reduce connection crossings. The case of the simple square topology (with a still smaller symmetry group) exhibits this pressure even stronger (Fig. 4.(a)).

In all runs the number of neurons in the best phenotype lie between 233-256 (maximum representable number of neurons). The algorithm thus efficiently exploits the possible range of neuron numbers.

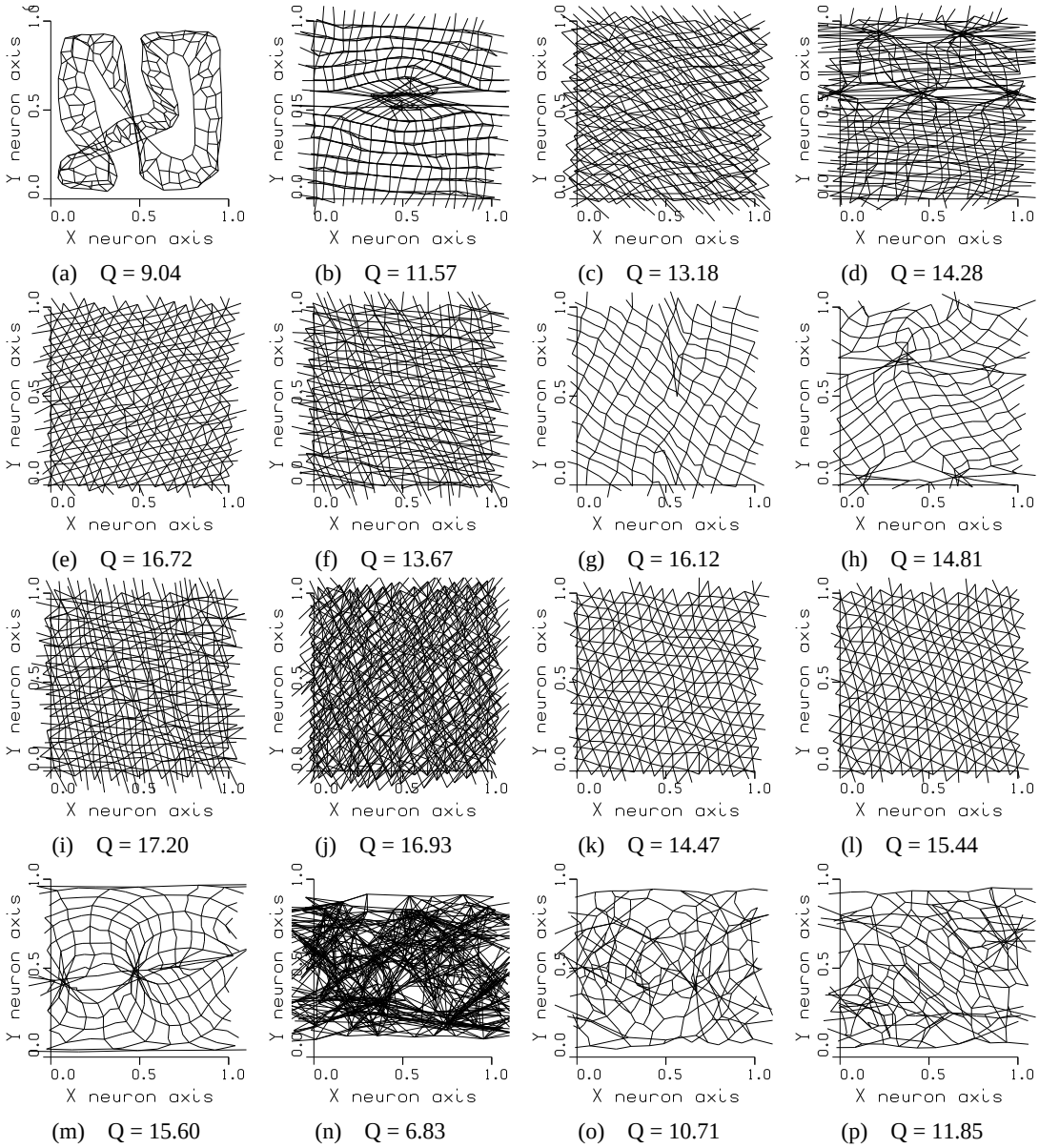

Figure 4: Phenotypes resulting from the GA runs.
Topology: (a) standard, (b)-(l) toroidal, (m)-(p) Möbius
Generations: (a) 500, (b) 0, (c) 11, (d) 1992, (e) 2440, (f) 213, (g) 3991, (h) 658, (i) 460, (j) 1175, (k) 454, (l) 415, (m) 1048, (n) 0, (o) 37, (p) 1777
Chromosome lengths: (b)-(i) and (k)-(m) use $n = 3$, (a) and (j) $n = 5$, (n)-(p) use $n = 7$
Activation profiles: (b)-(f), (i) use $h^{(1)}$, (a), (g)-(h), (j)-(p) use $h^{(2)}$
For more information see text.

6 CONCLUSIONS

We have applied GAs to create Kohonen Feature Maps able to adapt to a given input space without explicit assumptions on the topology needed. An earlier hypothesis that the transcription rule supports adaptation to highly symmetrical spaces has been confirmed. On the other hand, our expectations concerning the simulations using $h^{(2)}$ were not met. Neither the quality values were better nor were the net topologies more "flat" in a statistically significant way. Future work will concentrate on this question studying whether these hypotheses can be corroborated or must be rejected.

The concept of combining GAs and the Kohonen Feature Map paradigm exploits the particular given samples better than could be expected by theoretical considerations. The solutions found by the GAs might be used as a hint how the topologies of Kohonen nets should be chosen for faster convergence or adaptation to a given input space. In the current stage it is not clear whether a completely flat net topology gives a faster adaptation to a given input space than a net which is not completely flat (e.g. Fig. 4.(e)). Clarification of this question could yield a valuable tool for construction of net topologies for Kohonen-like models whenever speed of adaptation to a given input space is critical.

As a target of further research construction of transcription rules for different (and less symmetrical) topologies of input spaces is envisaged. As the transcription rule yields nets with a high degree of symmetry, a mechanism is therefore needed able to introduce a moderate reduction of symmetry needed to solve a given adaptation problem.

References

[1] Richard K. Belew, John McInerney, and Nicol N. Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. Technical Report CS 90-174, University of California, San Diego, 1990.

[2] Stefan Bornholdt and Dirk Graudenz. General asymmetric neural networks and structure design by genetic algorithms. Technical Report 91-046, Deutsches Elektronen-Synchrotron, Notkestraße 85, D-2000 Hamburg 52, May 1991.

[3] David J. Chalmers. The evolution of learning: an experiment in genetic connectionism. In D.S. Touretzky, J.L. Elman, T.J. Sejnowski, and G.E. Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*, San Mateo, California, 1990. Morgan Kaufmann.

[4] Bernd Fritzke. Growing cell structures - a self-organizing network in k dimensions. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks 2 - Proceedings of the 1992 International Conference on Artificial Neural Networks*, pages 1051–1056. Elsevier, 4-7 September 1992.

[5] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[6] John H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, 1975.

[7] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften.* Springer, Berlin, Heidelberg, New York, 1988.

[8] Teuvo Kohonen. *Self-Organization and Associative Memory*, volume 8 of *Springer Series in Information Sciences.* Springer-Verlag, Berlin, Heidelberg, New York, third edition, May 1989.

[9] Daniel Polani and Thomas Uthmann. Adaptation of Kohonen Feature Map Topologies by Genetic Algorithms. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, pages 421–429. Elsevier Science Publishers B.V., September 28-30 1992.

[10] Helge Ritter, Thomas Martinetz, and Klaus Schulten. *Neuronale Netze.* Addison-Wesley, 1990.

[11] Keiji Suzuki and Yukinori Kakazu. An approach to the analysis of the basins of the associative memory model using genetic algorithms. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the fourth international conference on Genetic Algorithms*, pages 539–546. Morgan Kaufmann, July 1991.

[12] Darrell Whitley and Thomas Hanson. Optimizing neural networks using faster, more accurate genetic search. In David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 391–396, San Mateo, California, 1989.